EUROPEAN PATENT SPECIFICATION

(12)

(54) A technique for producing an expert system for system fault diagnosis.

(73) Proprietor : AT & T Corp.
32 Avenue of the Americas
New York, NY 10013-2412 (US)

(72) Inventor : Lirov, Yuval V.
627 Randall Way
Aberdeen, New Jersey 07747 (US)
Inventor : Yue, On-Ching
57 Blevins Avenue
Middletown, New Jersey 07748 (US)

(74) Representative : Watts, Christopher Malcolm
Kelway, Dr. et al
AT&T (UK) LTD.
AT&T Intellectual Property Division
5 Mornington Road
Woodford Green Essex IG8 OTU (GB)

EP 0 410 632 B1

Description

Technical Field

5   The present invention relates to a computer-implemented technique for providing very fast and cost-efficient fault diagnosis systems which uses a hierarchy of rules in a knowledge base.

Description of the Prior Art

10   Testing methods for fault diagnosis in various circuits and systems have been recognized as a goal for many years. Early testing methods required a testing procedure to be devised and a tester to physically perform each test of the sequence and determine whether or not a fault occurred from the test step. In more recent years, some or all of the design of the testing procedure and the performance of actual testing steps have been automated using computer-based arrangements. In this regard see, for example, U.S. patents 4,228,537 and

15   4,242,751 issued to L. P. Henckels et al. on October 14, 1980, and December 30, 1980, respectively. In the earlier Henckels et al. patent, on-line simulation of circuit faults was used to generate a small part of a complete dictionary for a mini-computer based automated test system needed for diagnosing a circuit. This computer based test system includes only a small amount of secondary storage and is adapted for an exact match diagnosis with modeled failures, and a heuristic approach for a partial match of faulty behavior that lead to a

20   highly probable diagnosis. The later Henckels et al. patent discloses an automatic fault detecting computer system wherein external control is provided for introducing intelligence into the probing of circuit board nodes where insights into predictable or likely failures are available. More particularly, the look-ahead computer-guided probe makes use of a partial fault dictionary to supply predictions of fault location, based on the faulty behavior of the Unit Under Test (UUT). Instead of tracking the faulty behavior back from an edge connector of a

25   board, the system automatically instructs the operator to probe the point at which a fault is predicted.

    Expert systems have also been applied to diagnostic activities. In this regard see, for example, the article entitled "An Expert System For Help To Fault Diagnosis On VLSI Memories" by T. Viacroze et al. published in the Journal for the International Symposium For Testing and Failure Analysis (ISTFA), October-November 1988 at Los Angeles, California at pages 153-159. The disclosed system infers rules by taking into account

30   the many models of faults which can be found in a type of memory. A self documentable historical data base containing the conclusions of the electrical diagnosis performed by the Expert System, and the results of the technological analysis is used for statistical studies. The system first analyzes the data from the tester and infers the adequate rules and information available within the archive data base. The second level analysis by the computer is performed by a dialog with the operator. Finally, a fault assumption is generated, using prob-

35   ability coefficients, on a minimum functional block of the memory, and such assumption is verified providing forward and backward reasoning.

    In "Hybrid diagnostic strategy for an expert system controlled automatic test system (EXATS)", *AUTO-TESTCON '88 CONFERENCE PROCEEDINGS,* October 1988, pages 41-46, Minneapolis, USA, K.W. Pflueger discloses an expert control system that links an expert system to an Automatic Test System (ATS) for im-

40   proving the results of a sequence of tests that the ATS applies to a unit. In particular, the expert system determines which diagnosis (test program) is to be run on the ATS and then controls the sequence of tests that the test program applies to the unit under test. The expert system also communicates with an operator and learns from the failure history that results from such testing. The ATS, responsive to the program, applies test stimuli to the unit under test (UUT) and measures the result at various test points of the uut. The ATS then

45   supplies the result to the expert system, which then determines as a function of the result and failure history the next test stimuli that is to be applied to the UUT.

    When constructing expert diagnostic systems, one is usually concerned with the well-known "knowledge acquisition bottleneck", where the knowledge base is the heart of the expert system and involves the construction of the rules specific to the task domain. The rules of the diagnostic expert system produce directions to

50   a sensor-based system as to what, where and when to measure or replace in the system under test. One approach which can be adopted for dealing with this problem is based on a semantic control paradigm which is utilized to model the diagnostic/repair process, and the task of the semantic control procedure is to construct in real-time, the required relations, so that, when executed, the overall objective of the control process is achieved. The known procedures have produced long and expensive sequences, and the problem remaining in

55   the prior art is to provide a technique for the automated generation of efficient diagnostic procedures which include reasonably short and inexpensive sequences of measurement.

Summary of the Invention

The foregoing problem in the prior art has been solved in accordance with the present invention which relates to a computer-implemented technique for the automated generation of diagnostic procedures that use a particularly effective hierarchy of rules. More particularly, a first level of rules produces a decomposition of a system being diagnosed into groups of sequential and parallel subsections or subsystems. In this regard, a system is first decomposed into its main serially grouped sections, and then each main section which includes parallel connected components is further decomposed, if possible, into its major serial and parallel grouped subsections, which process reiterated until each section has been decomposed as much as possible into a serial string. At the second level are rules that generate the efficient testing rules for each pure subsection obtained from the decomposition process. The second level rules can be compared to a node evaluation function in a typical problem of searching a graph to select the best node for expansion from a current list of candidate nodes, so that the best path to the correct system diagnosis is found in the shortest amount of time using a minimum of user input. Two heuristic rules are applied by the second level rules to speed up the process of selecting the node as the best candidate for expansion to achieve a very cost efficient and fast fault detection system.

Other and further aspects of the present invention will become apparent during the course of the following description and by reference to the accompanying drawings.

Brief Description of the Drawings

FIG. 1 is a block diagram of a controller for practicing the present invention;

FIG. 2 is a block diagram of an exemplary hybrid circuit pack arrangement for illustrating the principles of the present invention; and

FIG. 3 is a test tree sequence for testing the circuit pack arrangement of FIG. 2 as derived in according with the rules of the present invention.

Detailed Description

The present invention relates to a technique for system fault diagnosis and, more particularly, to a technique for developing the knowledge base of any diagnostic expert system. As defined hereinbefore, the knowledge base usually consists of rules specific to the task domain and produce directions to a sensor-based system as to what, where, and when to measure or replace in the system under test. Efficiency of such a knowledge base is determined by the cost of the average test sequence. This cost is computed from the costs of measurements, failure rates of different components of the system, configuration or topology of the system, and the structure of the diagnostic rules. Currently used methods and computer programs for the automated generation of a knowledge base when given the above elements have produced inefficient knowledge bases resulting in a diagnostic expert system with unnecessarily long and expensive sequences of measurements.

Since circuit packs are the building blocks of all electronic equipments, the following discussion of the present invention will be directed to producing a knowledge base for a circuit pack which usually comprises a printed wiring board and components such as integrated circuits, resistors, capacitors, switches, and relays. It should be understood, however, that the present invention can also be used for providing knowledge bases and testing of other systems, and should not be limited to just circuit packs. As factories employ computed integrated manufacturing (CIM) techniques to (1) reduce work-in-progress (WIP) inventory, and (2) shorten manufacturing interval and improve quality; technicians responsible for diagnosing defective circuit packs should be able to troubleshoot any one from a large variety of products by adjusting the testing procedures to incorporate the latest component and process quality information, and learn to analyze new products quickly. In response to this increased demand for flexibility in the test and repair area, an expert system has been developed which suggests the tests to be performed to determine the cause of defect by consulting a database of circuit information and testing procedures and suggest which component(s) to replace.

As shown in FIG. 1, this expert system 10 consists of a graphical user interface 11 controlled by an inference engine 12 which is driven by a knowledge base 13. The expert system 10(1) accepts specifications of the diagnostic strategy and descriptions of the electronic unit under test (UUT) from a process control interface 14 and/or knowledge engineer interface 15 that uses a process control database as the basic design blocks, and (2) automatically transforms these specifications into compilable code (CP) 16 by using the knowledge base 13 about the UUT, programming structures and general diagnostic strategies. The present interface 11 is effectively a Graphical Diagnostic Expert System (GDES) which bridges between the expert system 10 and a team comprising a test engineer, circuit pack troubleshooter and development programmers which work via

interface 15 to construct and maintain the underlying circuit pack troubles hooting expert system. Interface 11 can be used to generate a database 13 of diagnostic rules, where the rule generation algorithm is based on the syntactic relations between the nodes of a graphical diagnostic tree. It is to be understood that the diagnostic knowledge base 13 can be developed in any suitable computer language as, for example, PROLOG, where the meta-programming techniques of PROLOG, coupled with object-oriented programming, all for efficient and fast implementation of the present expert system 10. Such meta-programming can create interactively the predicates and objects pertaining to the diagnostic reasoning, e.g., rules, advice, etc.

Specifically, the present Computer Aided Design (CAD) expert system uses a search of the knowledge base to aid the design of the troubleshooting computer programs which are to be used to assist the human troubleshooters. Such system must be able to develop a strategy given the description of the circuit and its behavior. The key to the present approach is to solve a test sequencing problem always for a purely sequential or a purely parallel case, since such approach allows for direct computation of the expected cost of the test sequence, thus allowing for its explicit minimization. To provide a technique for the automated generation of efficient diagnostic procedures which include reasonably short and inexpensive sequences or measurement, the present invention uses a particularly effective hierarchy of rules.

A first level of rules produces a decomposition of a system into groups of sequential and parallel subsections or subsystems. More particularly, the decomposition rules are as follows: '

Rule 1: if all components are connected in parallel then construct a single "par" node.

Rule 2: if all components are connected in sequence then construct a single "seq" node.

Rule 3: if all component sequences have a common prefix component then subdivide the "par (seq)" node into a "seq (par)" node of two "seq" nodes, one with the prefix component and the other comprising the rest of the components.

Rule 4: if several "par" subnodes of a common "seq" node have common suffixes/prefixes, then delete them from the "par" subnodes and insert them into the common "seq" node in the same order.

To first illustrate the first level of decomposition rules, consider the circuit pack arrangement of FIG. 2. The circuit pack arrangement of FIG. 2 comprises components 21-31 connected in various serial and parallel components with the input to the circuit pack being at 20 and the output at 32, and each component has an unweighted probability of failure as indicated above the component box. To appropriately decompose such arrangement, a small rule-based knowledge system is constructed in knowledge base 13 of FIG. 1 so as to provide the present expert system with the ability to make abstractions of the circuit pack of FIG. 2. Using this knowledge system, a general pattern (either sequential or parallel) is asserted initially. Then the components that do not fit into the chosen general pattern are aggregated into the "super component" level. More particularly, in FIG. 2 the circuit pack arrangement can initially be decomposed into it main serially grouped section including a serial arrangement of three boxes comprising a first box of component 21, a final box of component 31, and a central "super component" box of components 22-30 per Rules 1-3 above which can be designate as S1. A next decomposition step can start to decompose the central "super component" box using Rule 1-4 above. For example, the central super component box S1 of components 22-30 can be decomposed into two parallel boxes, with a first super component box including components 22,24 and 25 and designated S2, and a second super component box including the remaining components 23 and 26-30 and designated S3. A third decomposition step might decompose the first super component box S2 of components 22,24 and 25 into two sequential boxes with a first one of the boxes including component 22 and the second box in the series including the super component box of components 24 and 25 designated S4. It is to be understood that the remaining components of the second super component box S3 of components 23 and 26-30 can be decomposed in a similar manner where possible.

Having decomposed the circuit pack arrangement, four test tree generation rules are applied to such decomposed arrangement where a test tree is known in the art and is obtained by attaching circular nodes to the single links to provide a tree configuration of ambiguity sets and choice sets. The present test tree generation rules are:

Rule 1: if the node is a "par" node, then its cost "c" is computed from

$$c(node) = \min_{i} \{c(t_i) + q_i c(subnode_i)\}$$

where i runs over all possible partitions, $c(t_i)$ is the cost of performing test t_i, q_i is the $subnode_i$ probability, and the best test is determined from

$t_i = $ arg min c(node).

Rule 2: if the node is a "seq" node, then the cost "c" is computed from

$$c(\text{node}) = \min_{i} \{c(t_i) + p_i c(\text{leftsubnode}_i) + q_i c(\text{rightsubnode}_i)\}$$

where i runs over all possible partitions, left subnode corresponds to the node obtained when test t_i passed, right subnode when test t_i failed, and the best test is determined from

t_i = arg min c(node).

Rule 3: If the node is a "par" node, then its lower bound cost, the lowest cost possible for a node and also termed a cost underestimate, is computed from the exemplary steps:

Sort failure rates of components numerically in a first ascending or descending order into a list P.

Sort test costs of components numerically in a second descending or ascending order opposite the first order into a list L.

h ← 0

q ← 1

repeat the following steps:

select first member c from L

h ← h + c·q

select last member p from P

q ← q - p

until the number of elements in P=1.

The above rule includes steps where (1) failure rates of the various components of each subsystem are sorted numerically into a list P in a first ascending or descending order in the knowledge base; (2) the test costs of the same components are sorted numerically into a list L in the knowledge base in a second descending or ascending order which is opposite the first order; and (3) summing the product of each of the corresponding failure rates q and test costs c in lists P and L according to $\Sigma q_i c_i$. Such summing of the products can be performed by the variables h and q being initialized to 0 and 1, respectively, and a loop being performed where a first cost is selected from the list L, and the current value of variable h being added to the cost updated with the current value of variable q and stored as the new value of h, a last member of the list P being selected and the current value of the variable q being added to the current value of p and stored as the new value of q. The loop is reiterated until the list P only has one element remaining.

Rule 4: if the node is a "seq" node, then its lower bound cost, or cost underestimate, is computed from the steps of:

Sort failure rates numerically in a first ascending or descending order into P.

Sort test costs numerically in the first ascending or descending order into L.

h ← 0

repeat the following steps:

select first two members p_1 and p_2 from

P

p ← p_1 + p_2

select first member c from L

h ← h + p·c

insert p into P in an ordered form

until p = 1.

The sequence of Rule 4 can easily be determined from the explanation of the sequence provided for Rule 3. Having determined the lower bound costs, the diagnostic knowledge base is derived according to Rules 1 and 2. The above rule-base produces an efficient diagnostic knowledge base when used by a standard branch-and-bound search procedure. When the circuit pack arrangement of FIG. 2 is submitted as an input to the above designated procedure, the arrangement of FIG. 1 produces a diagnostic knowledge base test tree depicted in FIG. 3. From the test tree of FIG. 3 it can be seen that decomposition rules 1-4 were used to decompose the circuit pack arrangement of FIG. 2 as outlined hereinbefore, and that the test tree generation rules determined the points in the decomposed element where the sequence of tests should be performed to produce a fast and cost efficient fault diagnosis program. It is to be understood that there are many other diagnostic sequences that could be performed to test the circuit pack arrangement of FIG. 2, but the present invention produces a knowledge base that produces a fast and cost-efficient fault diagnosis program.

Claims

1. A method of forming a knowledge base in a computer for producing an expert system for the fault diagnosis of a predetermined arrangement of a system or device which comprises a plurality of components with separate predetermined failure rates, the method comprising the steps of:

   (a) decomposing the system or device into groups of sequential and parallel subsystems of one or more components each; and

   (b) generating a tree structure of the groups of step (a) by attaching nodes to each parallel and sequential link between subsystems in the tree to provide a tree configuration of suspected component ambiguity sets and possible measurement choice sets;

   CHARACTERIZED BY

   (c) computing a lower bound cost for each of the parallel and sequential subsystems using a first rule that (1) if a node is a parallel node, then its lower bound cost is computed by (i) sorting failure rates of the components of each subsystem numerically in a first ascending or descending order in a first list P, (ii) sorting a test cost of the components of each subsystem numerically in a second descending or ascending order, which is opposite in direction from the first order, in a second list L, and (iii) computing the product of each of the corresponding elements in lists P and L, and (2) a second rule that if the node is a sequential node, then its lower bound cost can be computed by (i) separately sorting each of the failure rate and the test cost for each component of each subsystem numerically in an ascending or descending order in the first and second list P and L, respectively, (ii) initializing a variable h to zero, (iii) selecting the lowest valued two numbers p_1 and p_2 from the list P, (iv) computing a current value for a failure rate p by summing p_1 and p_2 (v) selecting a first member c from list L, (vi) summing the current value of h with the product of the value of p and c from step (iv), and placing such sum for the current value for h, (vii) inserting the current value of p in numerical order in list P, and (viii) repeating steps (iii) to (vii) until p=1;

   (d) generating a diagnostic knowledge base including the cost of a parallel node and a sequential node for all possible partitions of the system under test, and

   (e) generating a diagnostic fault testing sequence at an output of the computer and as a function of the diagnostic knowledge base.

2. The method according to claim 1

   CHARACTERIZED BY

   in computing step (iii) for the lower bound cost for a parallel node performing the steps of:

   (c1) initializing a first and second variable h and q to 0 and 1, respectively;

   (c2) selecting a first member c from list L;

   (c3) computing the product of $h+c \cdot q$ and substituting such value for the current value of h;

   (c4) selecting a last member p from list P;

   (c5) subtracting the selected member p in step (c4) from the current value of q and substitute such subtracted value as the current value of q; and

   (c6) repeating steps (c2) to (c5) until the number of elements in list P=1.

3. The method according to claim 1

   CHARACTERIZED BY

   in performing step (d) performing the step of:

   (d1) computing the cost of a parallel node from the expression

   $$c(node) = \min_i \{c(t_i) + q_i c(subnode_i)\}$$

   where i includes all possible partitions of the system, q_i is the $subnode_i$ probability, and the best test is determined from

   $$t_i = \arg \min c(node).$$

4. The method according to claim 1

   CHARACTERIZED BY

   in performing step (d) performing the step of:

   (d1) computing the cost of a sequential node from the expression

   $$c(node) = \min_i \{c(t_i) + p_i c(leftsubnode_i) + q_i c(rightsubnode_i)\}$$

where i includes all possible partitions of the system, left subnode corresponds to the node obtained when test t_i passed, right subnode corresponds to the node obtained when test t_i failed, and the best test is determined from

$$t_i = \arg \min c \ (node).$$

5. A method according to any of the preceding claims
CHARACTERIZED BY
in performing steps (a) and (b), performing the steps of:
(e1) constructing a single parallel node if all of the components of the system are connected in parallel;
(e2) constructing a single sequential node if all of the components of the system are connected in sequence;
(e3) subdividing a parallel(sequence) node into a sequence(parallel) of a two sequence node, one with a prefix component and the other component including the remaining components, if all component sequences have a common prefix; and
(e4) if several parallel subnodes of a super component sequence node include common prefixes or suffices, then deleting such parallel subnodes and inserting them into the super component sequence node in the same order.

Patentansprüche

1. Verfahren zum Bilden einer Wissensbasis in einem Computer zum Erzeugen eines Expertensystems für die Fehlerdiagnose einer vorbestimmten Anordnung eines Systems oder eines Gerätes, welches mehrere Bauteile mit separaten, vorbestimmten Fehlerraten umfaßt, mit den folgenden Verfahrensschritten:
(a) Zerlegen des Systems oder Gerätes in Gruppen von sequentiellen und parallelen Untersystemen von jeweils einem oder mehreren Bauteilen, und
(b) Erzeugen einer Baumstruktur von den Gruppen nach Schritt (a) durch Verbinden von Knoten mit jeder parallelen und sequentiellen Verbindung zwischen den Untersystemen in dem Baum, um eine Baumkonfiguration von verdächtigten Bauteil-Mehrdeutigkeitssätzen und von möglichen Meßwahlsätzen bereitzustellen,
gekennzeichnet durch
(c) Berechnen einer niedrigeren Kostengrenze für jedes der parallelen und sequentiellen Untersysteme unter Anwendung einer ersten Regel, nach der, (1) wenn es sich bei dem Knoten um einen parallelen Knoten handelt, anschließend seine niedrigere Kostengrenze berechnet wird, indem (i) die Fehlerraten der Bauteile eines jeden Untersystems numerisch in einer ersten ansteigenden oder fallenden Reihenfolge in einer ersten Listen P sortiert werden, (ii) Prüfkosten für die Bauteile eines jeden Untersystems numerisch in einer zweiten fallenden oder ansteigenden Reihenfolge, deren Richtung entgegengesetzt der Richtung der ersten Reihenfolge ist, in einer zweiten Liste L sortiert werden, und (iii) indem das Produkt jedes der entsprechenden Elemente in den Listen P und L berechnet wird, sowie (2) unter Anwendung einer zweiten Regel, nach der, wenn es sich bei dem Knoten um einen sequentiellen Knoten handelt, anschließend die niedrigeren Kostengrenze dadurch berechnet werden kann, daß (i) jede Fehlerrate sowie die Prüfkosten für jedes Bauteil eines jeden Untersystems numerisch in einer ansteigenden oder fallenden Reihenfolge in der ersten und zweiten Liste P bzw. L getrennt sortiert wird, (ii) daß eine Variable h auf null gesetzt wird, daß (iii) der niedrigste Wert von zwei Zahlen p_1 und p_2 aus der Liste P ausgewählt wird, daß (iv) ein aktueller Wert für eine Fehlerrate p durch Summieren von p_1 und p_2 berechnet wird, daß (v) ein erstes Mitglied c aus der Liste L ausgewählt wird, daß (vi) der augenblickliche Wert von h mit dem Produkt des Wertes von p und c aus Schritt (iv) summiert wird, und daß diese Summe für den augenblicklichen Wert für h abgelegt wird, daß (vii) der augenblickliche Wert von p in numerischer Reihenfolge in die Liste P eingefügt wird, und daß (viii) die Schritte (iii) bis (vii) wiederholt werden, bis p=1;
(d) Erzeugen einer diagnostischen Wissensbasis mit den Kosten für einen parallelen Knoten und einen sequentiellen Knoten für alle möglichen Unterteilungen des zu prüfenden Systems, und
(e) Erzeugen einer diagnostischen Fehler-Prüffolge an einem Ausgang des Computers als Funktion der diagnostischen Wissensbasis.

2. Verfahren nach Anspruch 1,
dadurch gekennzeichnet,
daß im Berechnungsschritt (iii) für die niedrigere Kostengrenze für einen parallelen Knoten folgende

7

Schritte ausgeführt werden:

(c1) Setzen einer ersten und zweiten Variablen h und q auf 0 bzw. 1,

(c2) Auswählen eines ersten Elements c aus der Liste L,

(c3) Berechnen des Produkts von h + c x q und Ersetzen des augenblicklichen Werts von h durch diesen Wert,

(c4) Auswählen eines letzten Elements p aus der Liste P,

(c5) Subtrahieren des in Schritt (c4) ausgewählten Elements p von dem aktuellen Wert von q und Ersetzen dieses subtrahierten Wertes als den aktuellen Wert von q, und

(c6) Wiederholen der Schritte (c2) bis (c5) solange, bis die Anzahl von Elementen in der Liste P=1.

3. Verfahren nach Anspruch 1,
dadurch gekennzeichnet,
daß in dem Ausführungsschritt (d) folgender Schritt ausgeführt wird:

(d1) Berechnen der Kosten für einen parallelen Knoten aus der Gleichung

$$c(Knoten) = min_i\{c(t_i) + Q_i c(Unterknoten_i)\},$$

wobei i alle möglichen Unterteilungen des Systems enthält, q_i die Unterknoten$_i$-Wahrscheinlichkeit ist und die beste Prüfung aus

$$t_i = arg\ min\ c(Knoten)$$

ermittelt wird.

4. Verfahren nach Anspruch 1,
dadurch gekennzeichnet,
daß in dem Ausführungsschritt (d) der folgende Schritt ausgeführt wird:

(d1) Berechnen der Kosten für einen sequentiellen Knoten aus der Gleichung

$$c(Knoten) = min_i\{c(t_i) + p_i c(linker\ Unterknoten_i) + q_i c(rechter\ Unterknoten_i)\},$$

wobei i alle möglichen Unterteilungen des Systems enthält, der linke Unterknoten dem Knoten entspricht, der erhalten wird, wenn die Prüfung t_i erfolgreich war, wobei der rechte Unterknoten dem Knoten entspricht, der erhalten wird, wenn die Prüfung t_i fehlgeschlagen ist und wobei die beste Prüfung aus

$$t_i = arg\ min\ c(Knoten)$$

ermittelt wird.

5. Verfahren nach einem der Ansprüche 1 bis 4,
dadurch gekennzeichnet,
daß in dem Ausführungsschritt (a) und (b) die folgenden Schritte ausgeführt werden:

(e1) Bilden eines einzelnen parallelen Knotens, wenn alle Bauteile des Systems parallel geschaltet sind,

(e2) Bilden eines einzelnen sequentiellen Knotens, wenn alle Bauteile des Systems in Reihe miteinander geschaltet sind,

(e3) Unterteilen eines parallelen (sequentiellen) Knotens in eine sequentielle (parallele) Anordnung von zwei aufeinanderfolgenden Knoten, und zwar einer mit einem vorangestellten Bauteil, wobei das andere Bauteil die übrigen Bauteile aufweist, wenn alle Bauteilfolgen einen gemeinsamen Präfix besitzen, und

(e4) wenn mehrere parallele Unterknoten eines Superbauteil-Folgeknotens gemeinsame Präfixe oder Suffixe aufweisen, anschließendes Löschen solcher paralleler Unterknoten und Einfügen in den Superbauteil-Folgeknoten in der gleichen Reihenfolge.

Revendications

1. Méthode de formation d'une base de connaissances dans un ordinateur pour produire un système expert de diagnostic de défauts d'un dispositif prédéterminé d'un système ou périphérique qui comprend une pluralité de composants aux taux de défaillances prédéterminés distincts, la méthode comprenant les étapes de:

(a) décomposition du système ou périphérique en groupes de sous-systèmes séquentiels et parallèles d'un ou de plusieurs composants chacun; et

(b) génération d'une structure arborescente des groupes de l'étape (a) en reliant des noeuds à chaque liaison parallèle et séquentielle entre des sous-systèmes dans l'arborescence pour fournir une configuration arborescente d'ensembles d'ambiguités de composants suspectés et d'ensembles de choix

de mesures possibles;

CARACTERISEE PAR

(c) le calcul d'un coût limite inférieur pour chacun des sous-systèmes parallèles et séquentiels en uti-
lisant une première règle selon laquelle (1) si un noeud est un noeud parallèle, son coût limite inférieur
est alors calculé en (i) classant les taux de défaillance des composants de chaque sous-système nu-
mériquement dans un premier ordre croissant ou décroissant dans une première liste P, (ii) classant
un coût d'essai des composants de chaque sous-système numériquement dans un second ordre dé-
croissant ou croissant, qui est en sens opposé au premier ordre, dans une seconde liste L, et (iii) cal-
culant le produit de chacun des éléments correspondants dans les listes P et L, et (2) une seconde
règle selon laquelle si le noeud est un noeud séquentiel, son coût limite inférieur peut alors être calculé
en (i) classant séparément chaque taux de défaillance et chaque coût d'essai de chaque composant
de chaque sous-système numériquement dans un ordre croissant ou décroissant dans la première et
seconde liste P et L, respectivement, (ii) initialisant une variable h à zéro, (iii) sélectionnant les deux
nombres les plus bas p_1 et p_2 de la liste P, (iv) calculant une valeur actuelle d'un taux de défaillance p
en additionnant p_1 et p_2 (v) en sélectionnant un premier élément c de la liste L, (vi) additionnant la valeur
actuelle de h avec le produit de la valeur de p et c de l'étape (iv), et en plaçant cette somme comme
la valeur actuelle de h, (vii) insérant la valeur actuelle de p dans l'ordre numérique de la liste P, et (viii)
répétant les étapes (iii) à (vii) jusqu'à ce que p=1;

(d) la génération d'une base de connaissances de diagnostic comportant le coût d'un noeud parallèle
et d'un noeud séquentiel pour toutes les partitions possibles du système testé, et

(e) génération d'une séquence d'essais de défauts de diagnostic à une sortie de l'ordinateur et en fonc-
tion de la base de connaissances de diagnostic.

2. Méthode selon la revendication 1

CARACTERISEE PAR

lors de l'étape de calcul (iii) du coût limite inférieur d'un noeud parallèle l'exécution des étapes de:

(c1) initialisation d'une première et seconde variable h et q à 0 et 1, respectivement;

(c2) sélection d'un premier élément c de la liste L;

(c3) calcul du produit de $h+c.q$ et substitution de cette valeur comme valeur actuelle de h;

(c4) sélection d'un dernier élément p de la liste P;

(c5) soustraction de l'élément sélectionné p de l'étape (c4) à partir de la valeur actuelle de q et subs-
titution de cette valeur soustraite comme la valeur actuelle de q; et

(c6) répétition des étapes (c2) à (c5) jusqu'à ce que le nombre d'éléments dans la liste P=1.

3. Méthode selon la revendication 1

CARACTERISEE PAR

lors de l'exécution de l'étape (d) l'exécution de l'étape de:

(d1) calcul du coût d'un noeud parallèle à partir de l'expression

$$c(noeud) = \min_{i}\{c(t_i) + q_i c(sous\text{-}noeud_i)\}$$

où i comporte toutes les partitions possibles du système, q_i est la probabilité *sous-noeud_i* et le meilleur
essai est déterminé à partir de

t_i = arg min c *(noeud)*.

4. Méthode selon la revendication 1

CARACTERISEE PAR

lors de l'exécution de l'étape (d) l'exécution de l'étape de:

(d1) calcul du coût d'un noeud séquentiel à partir de l'expression

$$c(noeud) = \min_{i}\{c(t_i) + p_i c(sous\text{-}noeud\ gauche_i) + q_i c(sous\text{-}noeud\ droit_i)\}$$

où i comporte toutes les partitions possibles du système, le sous-noeud gauche correspond au noeud

obtenu quand l'essai t_i est réussi, le sous-noeud droit correspond au noeud obtenu quand l'essai t_i échoue, et le meilleur essai est déterminé à partir de

$t_i = $ arg min c *(noeud)*.

5. Méthode selon l'une quelconque des revendications précédentes
CARACTERISEE PAR
lors de l'exécution des étapes (a) et (b), l'exécution des étapes de:
(e1) construction d'un seul noeud parallèle si tous les composants du système sont connectés en parallèle;
(e2) construction d'un seul noeud séquentiel si tous les composants du système sont connectés en série;
(e3) sous-division d'un noeud parallèle (séquentiel) en un noeud séquentiel (parallèle) d'un noeud à deux séquences, l'un avec un composant préfixe et l'autre composant comportant les autres composants, si toutes les séquences de composants ont un préfixe commun; et
(e4) si plusieurs sous-noeuds parallèles d'un noeud séquentiel à super composants comporte des préfixes ou suffixes communs, suppression alors de tels sous-noeuds parallèles et insertion de ceux-ci dans le noeud séquentiel à super composants dans le même ordre.

FIG.1

REPLACEMENT MEASUREMENT → COMPILABLE PROGRAM → (16)

TEST ENGINEER DESIGNER

PROCESS CONTROL DATABASE

(10)

GRAPHICAL USER INTERFACE (11)

INFERENCE ENGINE (12)

KNOWLEDGE BASE (13)

PROCESS CONTROL INTERFACE (15)
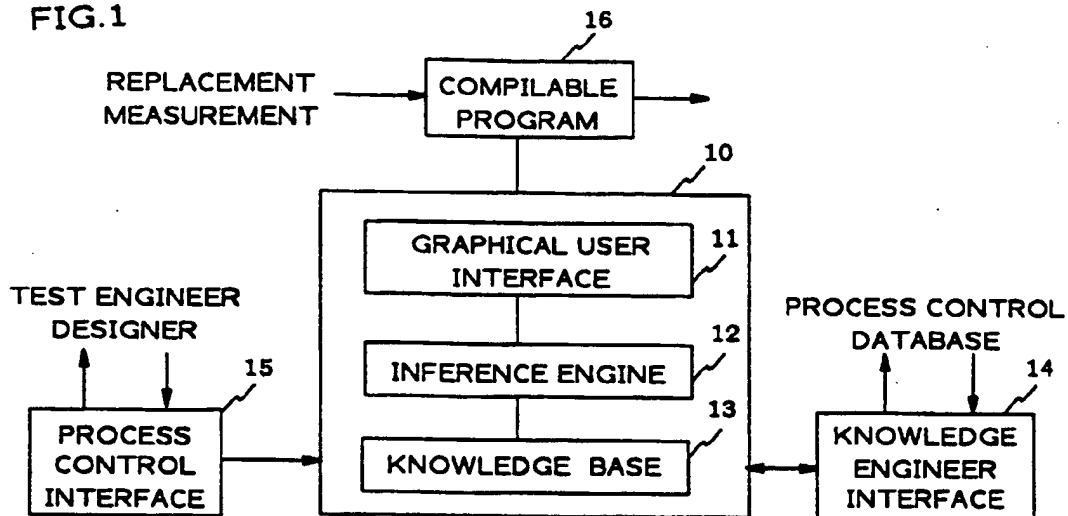
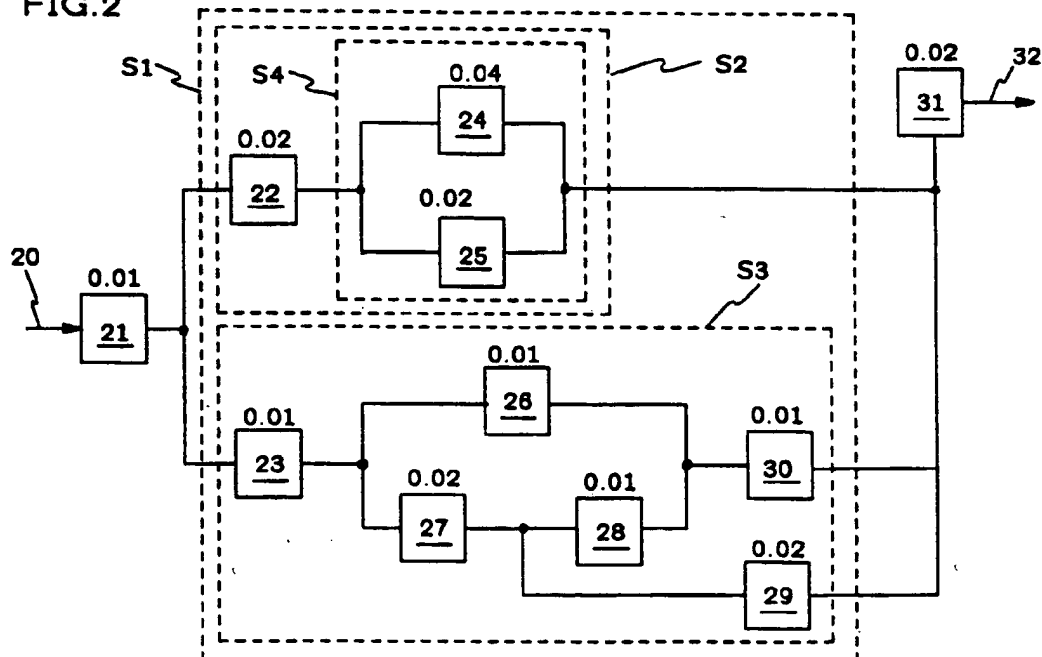KNOWLEDGE ENGINEER INTERFACE (14)

FIG.2

FIG.3

```
(* at the highest hierarchy level, the circuit consists of *)
(* components 21, S1, and 31 in sequence                   *)
measure input to 31;
if passed then check 31
else begin
     measure output from 21;
     if failed then check 21
     else begin (* troubleshooting S1, which consists of S2 and S3   *)
                (*                                      in parallel   *)
                (* first is S2 which consists of 22 and S4 in sequence *)
     measure output of 24 and 25;
     if failed then begin
           measure output of 22;
           if failed then check 22
           else begin (* S4 consists of 24 and 25 in parallel *)
               measure output of 25;
               if failed then check 25
               else check 24
           end
           else begin (* next is S3 *)
               measure output of 27;
               if failed then begin
                   measure output of 23;
                   if failed then check 23
                   else check 27
               end
               else begin
                   measure output of 30;
                   if failed then begin
                       measure input to 30
                       if failed then begin
                           measure output of 26;
                           if failed then check 26
                           else check 28
                       end
                       else check 30
                   end
                  else check 29
               end
     end
end
```